The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.

TLP: WHITE
Disclosure is not limited. Subject to standard copyright rules, TLP: WHITE information may be distributed without restriction.
http://www.us-cert.gov/tlp/

DATE(S) ISSUED:
01/19/2017

SUBJECT:
Multiple Vulnerabilities in PHP Could Allow for Arbitrary Code Execution

OVERVIEW:
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow an attacker to execute arbitrary code. PHP is a programming language originally designed for use in web-based applications with HTML content. PHP supports a wide variety of platforms and is used by numerous web-based software applications. Successfully exploiting the most severe of these vulnerabilities could allow for remote attackers to execute arbitrary code in the context of the affected application. Failed exploitation could result in a denial-of-service condition.

THREAT INTELLIGENCE:
There are currently no reports of these vulnerabilities being exploited in the wild. There is known proof-of-concept code for these vulnerabilities.

SYSTEMS AFFECTED:
- PHP 7.0 prior to 7.0.15
- PHP 7.1 prior to 7.1.1

RISK:
Government:
- Large and medium government entities: **High**
- Small government: **High**

Businesses:
- Large and medium business entities: **High**
- Small business entities: **High**

Home users: Low

TECHNICAL SUMMARY:
PHP has released updates that address multiple vulnerabilities, the most severe of which could allow for arbitrary code execution. These vulnerabilities include:

Prior to 7.0.15
- Bug #72555 (CLI output(japanese) on Windows).
- Bug #73646 (mb_ereg_search_init null pointer dereference).

- Bug #73654 (Segmentation fault in zend_call_function).
- Bug #73668 ("SIGFPE Arithmetic exception" in opcache when divide by minus 1).
- Bug #73686 (Adding settype()ed values to ArrayObject results in references).
- Bug #73727 (ZEND_MM_BITSET_LEN is "undefined symbol" in zend_bitset.h).
- Bug #73746 (Method that returns string returns UNKNOWN:0 instead).
- Bug #73783 (SIG_IGN doesn't work when Zend Signals is enabled).
- Bug #73789 (Strange behavior of class constants in switch/case block).
- Bug #73794 (Crash (out of memory) when using run and command separator).
- Bug #73847 (Recursion when a variable is redefined as array).

Prior to 7.1.1

- Bug #46103 (ReflectionObject memory leak).
- Bug #69425 (Use After Free in unserialize()).
- Bug #70513 (GMP Deserialization Type Confusion Vulnerability).
- Bug #72731 (Type Confusion in Object Deserialization).
- Bug #73092 (Unserialize use-after-free when resizing object's properties hash table).
- Bug #73585 (Logging of "Internal Zend error - Missing class information" missing class name).
- Bug #73586 (php_user_filter::$stream is not set to the stream the filter is working on).
- Bug #73612 (preg_*() may leak memory).
- Bug #73615 (phpdbg without option never load .phpdbginit at startup).
- Bug #73764 (Crash while loading hostile phar archive).
- Bug #73768 (Memory corruption when loading hostile phar).
- Bug #73773 (Seg fault when loading hostile phar).

Prior to 7.0.15 and 7.1.1

- Bug #31875 (get_defined_functions additional param to exclude disabled functions).
- Bug #67474 (getElementsByTagNameNS filter on default ns).
- Bug #70213 (Unserialize context shared on double class lookup).
- Bug #70490 (get_browser function is very slow).
- Bug #72931 (PDO_FIREBIRD with Firebird 3.0 not work on returning statement).
- Bug #73154 (serialize object with __sleep function crash).
- Bug #73265 (Loading browscap.ini at startup causes high memory usage).
- Bug #73373 (deflate_add does not verify that output was not truncated).
- Bug #73462 (Persistent connections don't set $connect_errno).
- Bug #73594 (dns_get_record does not populate $additional out parameter).
- Bug #73663 ("Invalid opcode 65/16/8" occurs with a variable created with list()).
- Bug #73679 (DOTNET read access violation using invalid codepage).
- Bug #73704 (phpdbg shows the wrong line in files with shebang).
- Bug #73737 (FPE when parsing a tag format).
- Bug #73753 (unserialized array pointer not advancing).
- Bug #73792 (invalid foreach loop hangs script).
- Bug #73800 (sporadic segfault with MYSQLI_OPT_INT_AND_FLOAT_NATIVE).
- Bug #73825 (Heap out of bounds read on unserialize in finish_nested_data()).
- Bug #73831 (NULL Pointer Dereference while unserialize php object).
- Bug #73832 (Use of uninitialized memory in unserialize()).
- Bug #73868 (DOS vulnerability in gdImageCreateFromGd2Ctx()).
- Bug #73869 (Signed Integer Overflow gd_io.c).

Successfully exploiting the most severe of these vulnerabilities could allow for remote attackers to execute arbitrary code in the context of the affected application. Failed exploitation could result in a denial-of-service condition.

RECOMMENDATIONS:
The following actions should be taken:

- Upgrade to the latest version of PHP immediately, after appropriate testing.
- Verify no unauthorized system modifications have occurred on system before applying patch.
- Apply the principle of Least Privilege to all systems and services.
- Remind users not to visit websites or follow links provided by unknown or untrusted sources.

REFERENCES:
NOTE: Visiting these links may trigger an IDS signature match for a Possible Encrypted Webshell Download. This is a false positive alert that is matching content on the pages below.

PHP:
http://php.net/ChangeLog-7.php#7.0.15
http://php.net/ChangeLog-7.php#7.1.1